

# Synthesizing Concepts in self-modification

Simon Hickinbotham and Susan Stepney

Department of Computer Science, University of York, UK  
York Centre for Complex Systems Analysis  
email: sjh518@york.ac.uk

## Introduction: What topics will the review address?

This proposal attempts to review and to unify concepts of self-reference and self-modification in living and artificial systems.

The idea that a system can be specified in abstract terms is common to the Biological and Computer Sciences. All living organisms have their phenotype encoded in DNA and (almost) all running computer code exists as source code. Both DNA sequences and computer programs can be thought of as abstract descriptions of the running system. However, we must recognise that the abstract description can never be complete, apart from the ridiculous case where the system is represented by an exact copy of itself, which is at best inefficient and at worst impossible. DNA offers a *partial* coding of the phenotype – the locally prevailing physics and chemistry provide the rest. In a similar way, computer programs offer a partial coding for the organisation of CPU processes – they way the computational framework compiles and runs the program does the rest.

In both of these systems, the coding is not only incomplete, but it is interpreted by another entity – it is by no means the only possible representation, and it can only be transcribed into the executing system if it is read by an appropriate interpreter. A special property of biological systems and some computer programs is that the interpreter itself can be encoded in the abstract representation (given the incompleteness of the representation we describe above). This closed system of interpretation has a property defined by Pattee as *semantic closure* [3] and it has a number of features that merit further study. In particular, it allows for the code base to be interpreted in different ways, depending on the structure and function of the entity that interprets it.

## Which subfields will the review bring together?

**Evolutionary Biology** Firstly, let's consider the evolved coding system in biology. It is well known that the codon table of DNA was fixed very early in the history of life and is very rarely deviated from. There are a number of ways

in which the coding table is optimal, and it is obvious that this optimisation is at the heart of biological evolution, but it has also itself evolved – and this is without considering the manner in which the nucleic and amino acids have special properties that make them ideal for the semantic closure relationship described above.

Research in this area is linked to research at the origin of life. There is much evidence as to the chemical basis for the coding table - with the first letter in the coding triplet corresponding to the amino acid precursor and the second corresponding to the hydrophobicity of the result. It is only the third codon that has a computational property, building redundancy into the system that allows neutral drift in the coding pattern.

**Computational reflection** Secondly, let's look at how semantic closure might work in computer science. Computational reflection uses software architectures that are capable of self-modification at runtime. These systems have implementations between two extremes: procedural reflection, in which unlimited self-modification is available at the expense of infinite recursion; and declarative reflection, which uses pre-defined metrics to drive the self-modification and is hence limited in scope. The authors have a paper at the main conference [2] which unifies the concept of computational reflection with biological systems via a new analysis of von Neumann's Universal Constructor (see figure 1). The result is a bio-reflective architecture that is capable of unconstrained self-modification without the problems of infinite recursion that exist in the computational counterparts. The new architecture is a blueprint for applications in Artificial Life studies, Evolutionary Algorithms, and Artificial Intelligence.

**Chemistry: Synthetic systems** Thirdly, let's think about how we might generate semantically closed systems *ab initio* using chemistry. An active area of research in chemistry is how to build self-governing chemical systems that produce sophisticated chemical systems more reliably and cheaply than industrial processes [1]. A challenge here is to construct controlling sub-reactive systems which intimately

control the local reactive processes. What we are looking for here is evidence of symbolic manipulation, but ideally we want to do it without knowing what the symbols are, or how they are embodied. Essentially this area is looking at the concept of the emergence of coding languages in complex systems.

### How will synthesizing these bodies of research benefit the field(s)?

Our aim is to foster the emergence of new thinking about how to build systems that have evolvable semantic closure at their core.

One of the big problems with studying biological systems in isolation from other disciplines is that we've only got one fossil record, and it is incomplete. It makes it difficult to discover why things are the way they are – be they frozen accident of functional necessity. Unifying a review of the way semantic closure can work across disciplines will allow biologists to uncover functional necessity in what has previously been thought of as frozen accident.

There is a similar frozen accident at the heart of computer science. This is not because it has evolved of course, but rather because computers are complicated things, and standards and methods have arisen to make computers work in the way we expect them to. This is what Ackley calls the 'robust first attractor'. It is basically the von Neumann 'Princeton' architecture (not his Universal Constructor architecture) and has the following features: emphasis on serial programming; demand for the exact; abstract representation. Our conjecture here is that some of these features present barriers to the creation of artificial living systems, in particular the emergence of semantically closed systems.

But we have another problem, and it is what von Neumann called the 'important half' of the problem. This is to do with discovering which abstract representation is most appropriate for specifying the system at hand. Too coarse, and the abstract representation becomes inflexible. Too fine-grained, and the number of parts becomes too great and the machine is too unwieldy. Somewhere in between is a 'sweet spot' of representation, within which the granularity is correct and the self-reference is the most appropriate. Semantic closure somewhere near to this level of representation should allow the system to go on to manage and refine its own representation. All we have to do is find a representation that falls within this attractor.

One way to break out of this thinking is to look at chemical self-organisation. Reviewing work on how self-organisation is controlled in chemical systems could reveal new ways of fostering semantic closure in biological and computational systems. Things such as the role of entropy, spatial organisation and the propagation of local reactions could all have an impact on how reference to a coding language might prove beneficial for the purpose of maintaining longer term state.

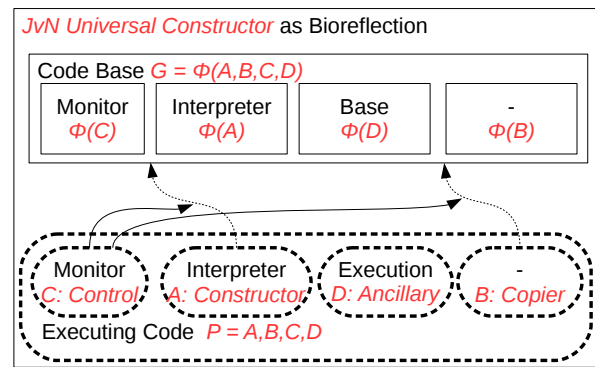


Figure 1: Bio-reflective Architecture. Universal Constructor terms are shown in red, from [2].

### Examples of terms and/or concepts used differently across the fields

We give here three inter-related terms that cover the concepts described above. Figure 1 shows how reflection and semantic closure are unified in the universal constructor architecture.

**Reflection:** This is computation which is 'about itself', and which allows the system to bring about changes in its own implementation. Our central theme is that of *reflection as a definition of life*. Reflection is defined as computing which is *about itself*, and in the same way, a 'living system' is also about itself.

**Semantic closure:** A phrase coined by Howard Pattee to describe the relationship between the physical and symbolic representations of a system.

**Universal constructor** An abstract machine architecture presented in the groundbreaking work of John von Neumann in the late 1940s: it specifies a set of classes of machine 'sub-assemblies' that together are capable of creating or evolving any other machine of arbitrary complexity.

### References

- [1] Alon Henson, Juan Manuel Parrilla Gutierrez, Trevor Hinkley, Soichiro Tsuda, and Leroy Cronin. Towards heterotic computing with droplets in a fully automated droplet-maker platform. *Phil. Trans. R. Soc. A*, 373(2046):20140221, 2015.
- [2] Simon Hickinbotham and Susan Stepney. Bio-reflective architectures for evolutionary innovation. In *ALife XV*. MIT Press, 2016.
- [3] Howard H Pattee. Cell psychology: an evolutionary approach to the symbol-matter problem. *Cognition and Brain Theory*, 5(4):325–341, 1982.